# CORE: Conservative Remapper

*Martin Staley and Mikhail Shashkov (T-7)*

Conservative Remapper, or CORE, is a C++ language software library for remapping cell masses and cell-averaged densities on unstructured two-dimensional grids, maintaining conservation of total mass in the process. CORE contains implementations of two remapping algorithms: a new, efficient "swept-region" algorithm and a more traditional algorithm based on the computation of cell intersections. Grids may be Cartesian or cylindrical, and cells may have three or more vertices, with no upper limit. CORE can run in serial and in parallel, but in order to achieve wide applicability, CORE uses no particular parallel communication library. Instead, it achieves parallel communication through strategically placed, user-defined callbacks. Users can also provide callbacks to redefine different parts, or subcomponents, of the remapping process. CORE allows the use of different data types, e.g., single, double, and quadruple precision floating-point numbers, through the use of C++ templates. Using CORE is simple and requires no configuration scripts or makefiles.

The swept-region remapping algorithm [1] achieves its goal by performing three steps: density reconstruction, mass exchange, and mass repair. This algorithm performs well if the new grid is a small perturbation of the old grid, and it is faster than the exact-intersection algorithm.

For the density reconstruction step, consider that the mean cell densities obey some underlying, theoretical density function $\rho$. For the reconstruction step, we assume $\rho$ is piecewise linear—one piece per cell—and then use our discrete mean densities to reconstruct $\rho$.

For the mass exchange step, consider that each edge has two adjacent cells. For a given edge, we identify which of these cells the edge "moves into" more when we go from the old grid to the new grid. We then integrate the cell's reconstructed density function over the entire region swept by the edge's motion. This gives us a mass, which is then removed from this cell and added to the cell on the other side of the edge.

Consider Fig. 1, where the old grid is drawn with dotted lines and the new grid with solid lines. When the edge illustrated in the figure moves from its original position to its final position, it sweeps the shaded region. So, we integrate Cell 5's reconstructed linear density function over this region, giving a mass which is subtracted from Cell 5 and added to Cell 4.

For the mass repair step we recognize that the earlier, mass exchange step involved inexact integration, in that only one cell's piece of the reconstructed density function (Cell 5's piece in the figure) was considered, even though portions of the swept region intersect with other cells. Because of inexact integration, new masses in individual cells can conceivably violate local bounds. For example, a mass might be negative. The repair step fixes out-of-bounds masses, while conserving the total mass over the entire grid.

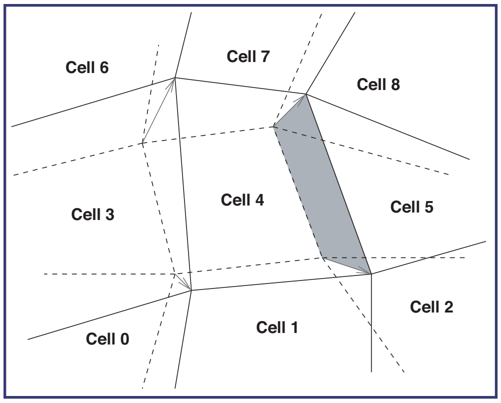

***Figure 1— Swept-region remapping.***

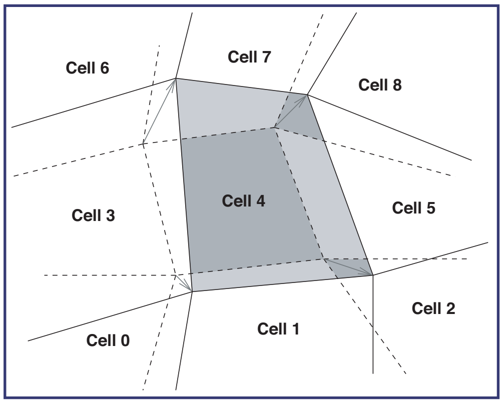

***Figure 2— Exact-intersection remapping.***

The exact-intersection remapping algorithm [2] begins with the same density reconstruction step used in the swept-region algorithm. However, the mass exchange step is now quite different. Consider Fig. 2, in which Cell 4 moves from its position on the old grid (dotted lines) to its position on the new grid (solid lines) as shown by the arrows. Exact-intersection remapping examines each of the grid's cells. Cell 4 on the new grid intersects with Cells 1, 2, 4, 5, 7, and 8 on the old grid, as shown by the shaded regions in the figure. So, the algorithm computes Cell 4's new mass by summing the integral of Cell 1's reconstructed density function over Cell 4's intersection with old Cell 1, the integral of Cell 2's reconstructed density function over Cell 4's intersection with old Cell 2, etc.

Although this algorithm no longer involves inexact integration, as the swept-region algorithm did, it still allows for a mass repair step in the event that masses fall outside user-defined bounds.

The exact-intersection algorithm does not require that the new grid is a small perturbation of the old grid. Moreover, it allows the two grids to have different connectivities, whereas the swept-region algorithm only makes sense when the grids have the same connectivity. Figures 3 and 4 illustrate this fact with a remap from a coarser grid to a much finer grid. Notice how the original grid's relative coarseness is still visible in the remapped values on the finer grid.

[1] M. Kucharik, M. Shashkov, and B. Wendroff, "An Efficient Linearity-and-Bound-Preserving Remapping Method," *J. Comp. Phys.*, **188** (2003) pp. 462–471.
[2] L. Margolin and M. Shashkov, "Second-Order Sign-Preserving Conservative Interpolation (Remapping) on General Grids," *J. Comp. Phys.*, **184** (2003) pp. 266–298.

***For more information, contact Martin Staley (mstaley@lanl.gov).***

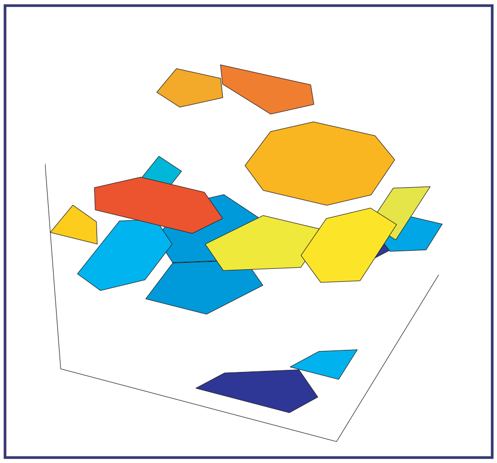

***Figure 3—* Densities on the original, coarser grid.**

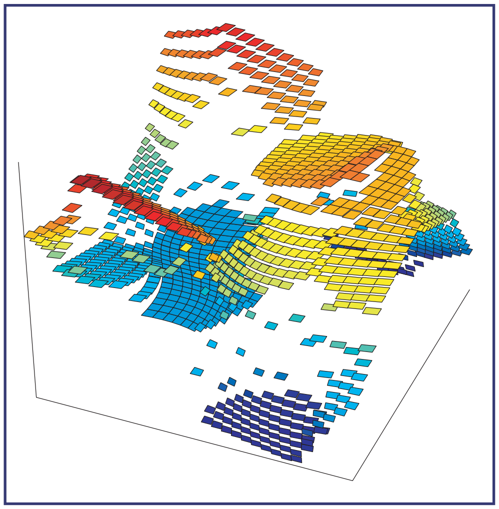

***Figure 4—* Remapped densities on the new, finer grid.**

**Acknowledgements**
We would like to acknowledge NNSA's Advanced Simulation and Computing (ASC), Advanced Applications Program, for financial support.